

ASIC 和 FPGA 设计中的多点综合技术

即使在今天 Moore 定律仍然有效。尽管在技术发展的每一个时刻做出精确的预言是困难的，但 ASIC 和 FPGA 所集成的门数仍象数年前 Intel 的 Gordon Moore 预言的那样平均每 18 个月增加一倍。事实上，通过长时间的观察这种在单个芯片上集成越来越多的功能的趋势，一些人深信这种集成的趋势加快了。今天的片上系统 (SoC) 一般包括了数字逻辑，存储器，微处理器，混合信号等模块，在不久的将来嵌入式的 FPGA 和模拟电路也会集成在单个芯片上。而且，今天的 IC 设计都是以硅的工艺作为基础的，今天的设计工程师有很多的选择在采用 ASIC 还是采用可编程逻辑器件，这一意味着 SoC 的设计将沿着 ASIC 和可编程逻辑器件两个方向发展。

像我们中的大多数人所熟知的那样，目前大多数的 IC 的实现都是建立在硅的工艺上。一方面，我们目前的工艺技术可使我们实现许多复杂的设计。另一方面，由于设计工具的局限使我们不能把工艺技术的许多潜在的能力发挥出来。面积和设计的复杂性成为在设计 IC 中的主要限制的因素。这种所谓的“生产率瓶颈”(图 1)正在不断的扩大，尽管我们专注于 EDA 工具大约已有十年之久。

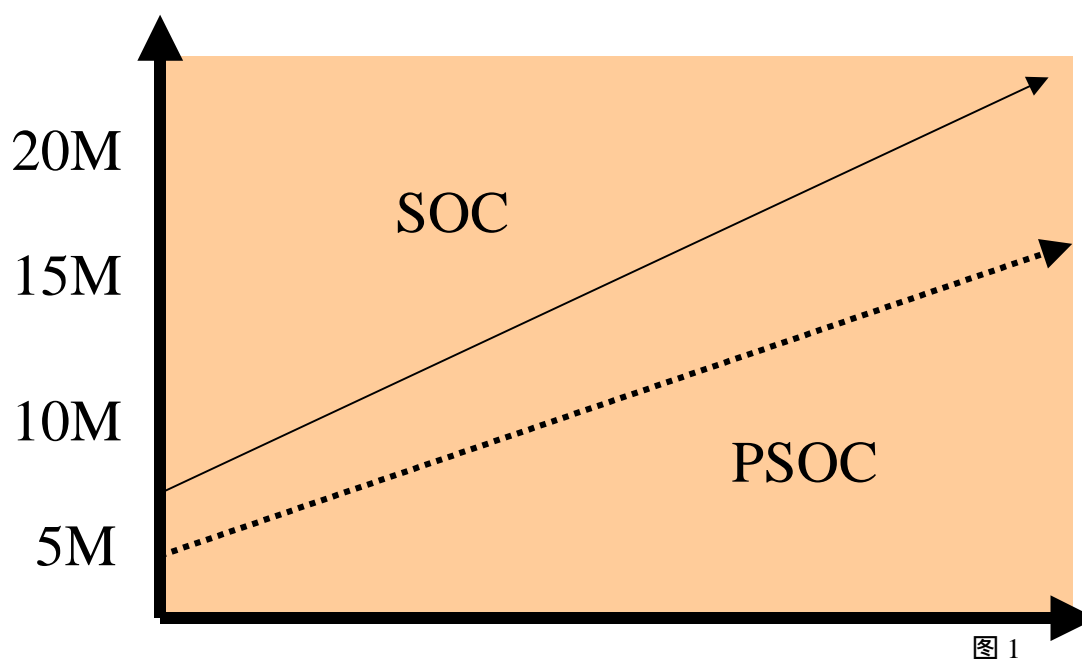


图 1

综合技术面临的挑战

没有任何地方比在综合的领域“生产率瓶颈”表现更为明显的了。综合工具本身能力限制了在做综合时，子模块的门数不能超过 200K。随着 200 万门设计的出现，工程师不得不把一个设计分成 100 或更多的子模块来进行综合。另外，工程师更习惯按照功能和时序来进行子模块的划分，而不是按照设计门数的大小。另外，没有单独的对独立功能的模块进行很好的优化，从而导致综合的效果也不好。例如在综合时应独立的对数据路径进行优化。嵌入式的 FPGA 应当在整个芯片级进行映射和优化。

假如在设计复杂性的缓慢增长是因为所采用的综合技术仍是来源于 10 年前的技术，而没

有进行大的技术突破，那么这种传统的设计方法的效率是令人怀疑的。由于设计的复杂性，在综合过程中产生了很多的问题。内存的利用在综合的过程中是一个巨大的挑战，因为一个精密的综合过程需要处理大量的数据。综合的时间也将变得很长，这不仅是因为设计的规模变大了，而且在顶层的设计管理也要处理大量的计算任务。IP 的集成也是个巨大的挑战，当在综合复杂的，有时需要的约束条件又是很严格的设计的时候，由于需要的反复的迭代来满足时序条件，或者内部模块设计的时间变长导致了复杂设计的周期变长。

对综合的另外的挑战是保持综合的稳定性，也就是说在一定的范围内对设计做增量综合时，而综合的结果却没太大的变化。在增量综合中，对设计的一部分重新定义，然后把重定义后的子模块放在整个的系统中时却产生时序的混乱。这种时序上的混乱随着设计复杂性的增长而增长。一般的，增量综合不能完全的解决模块外部的行为，这就造成了在接下来的设计中，已经优化的模块和设计的其他的部分的连接将产生紊乱。如果这种改变对设计的很多的模块都有影响的话，那么增量综合也使设计的形式验证变的很困难。

可选的设计方法和技术

为了避免综合工具的缺点，设计工程师应当选择多种解决方案。一种方案是利用精密的脚本文件。这是一种比较详细的，聪明的解决方案，但常常这种方案要在综合的质量(QoR)和综合的效率之间进行折衷，另外设计工程师还需要专业的技术。通常成功的综合一个大型的 ASIC 需要手工编写脚本文件，但是这种手工编写的脚本文件容易发生错误，这就需要工程师有专业的脚本编写技术。这种努力也许对设计没有太大的价值，花费的人力和时间完全可以用于提高设计的结果。综合工具对内存的要求限制了综合的门数，也削弱了脚本文件的作用。

另外一种方案是专注于 RTL 级的设计，而把综合和具体的物理实现交给 ASIC vendor。设计工程师避免了单调的综合过程，而把这种工作交给 Silicon vendor 去完成。这种“RTL signoff”（做完 RTL 级的仿真就把其他的工作交给专门的公司完成）方式在很大程度上依靠高质量的 RTL 级的代码，以及提供这种服务的公司愿意为设计的结果承担责任。设计工程师也许会担心把一个设计交给一个对这个设计和系统都缺乏了解的人，从而使设计的成功变得不可预测。

这些方案，都是假定设计不可能在被优化和没有瓶颈的限制。然而，综合是在 IC 的流程中所必需的。不管是谁进行综合，在现有的技术下，都要对综合的质量和综合的效率进行折衷。因此，综合问题必须得到关注。

Top-Down vs. Bottom-Up

通常在综合过程有两种方法，自顶向下和自底向上的方法。每一种方法都有它的优缺点。使用自底向上的方法的时候，要把设计划分为多个模块，并且每个模块的大小都可为综合工具所接受。通过把一个设计分为多个部分，每一个都可进行单独的综合并且按照设计的需要进行改变。这种方法允许对设计的一部分进行编译和处理，从而大大加快了编译的速度。自顶向下的方法也允许把设计的一部分独立出来进一步的进行综合和优化，但这要以不影响设计的其它部分为前提。如果操作正确的话，这将大大的提高设计结果的稳定性。

自底向上的方法对每一个模块的运行时间最短，但整个的运行时间长的令人难以忍受，因为要考虑到项目的管理和手工脚本的时间。在写脚本上的努力是显著的，但这样也容易产生错误。另外，也可能为 QoR（综合的质量）付出代价。用自底向上的方法，仅可以对模块的内部进行综合，而不能对模块的边界部分也进行综合。在整个设计中，将失去了跨模块

综合的可能性。随着边界的增加，设计将越来越远离最佳的综合结果。两个模块的时候你失去了一些优化设计的机会，当有 100 个模块的时候，你将失去的更多。

自顶向下的方法可以利用系统级的 RTL 代码和约束条件进行综合，在一个操作内，完成设计的优化并且得到门级网表，而不必要进行模块的划分。与自底向上的方法向比较，这种综合的方法得到的综合结果最好，由于在整个的设计上进行综合，手工编写脚本的量也减少了，也不必管理许多的模块，因此做起来也比较容易。然而，对于规模较大的设计，内存和运行的时间是综合的限制条件。设计中即使只有微小的变化都要重新进行综合，除非对比较小的设计，否则这是不切实际的。另外，当设计中有许多的重复的模块的时候，自顶向下的方法要把这个模块拷贝多次进行综合，而不象自底向上的方法那样只拷贝一次，从而设计的时间变得很长。最后一点，自动布局布线能够以流水线的方式和迭代的方式来满足时序的要求，如果一个设计中有相同的模块，以层次化的方法进行综合，又将用层次化的方法进行布局布线。自顶向下的方法对综合的质量是比较理想的，但考虑到设计的一些需要和设计的规模，也许并不是对每一个设计都是很好的方法。

多点综合——大型设计的综合技术

许多的设计都是采用即不是完全的自顶向上也不是纯粹的自底向上的方法。作为一个可替代的方法，把自顶向下方法的易使用、高的综合质量结果与自底向上方法的高效率、低的内存使用结合起来，这是客户最需要的。在规模大的设计中遵循着高质量的综合结果和较少的运行时间，从而在哪儿采用自顶向下，哪儿采用自底向上的方法是极其的重要的。为了解决设计工程师在设计百万门级的 ASIC、SoC 和 PSoC 所面临的挑战，Synplicity 公司开发了多点综合的技术。

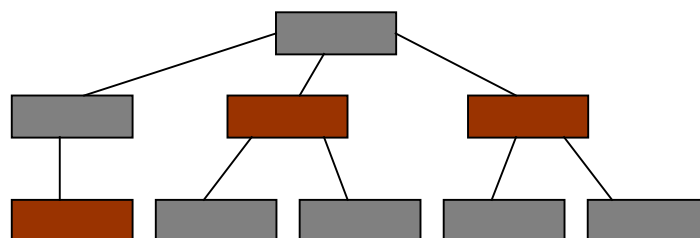
多点综合技术在大型的设计中利用了层次化的方法。与通常的自底向上的设计方法相比较，多点综合技术可以自动的划分和优化设计。在对整个的设计完成编译后，产生了一个包含有层次化信息和其他的关于设计的关键信息的中间格式的文件。用户可利用 Synplify ASIC 内嵌的 HDL Analyst 工具去观察 RTL 设计的层次，也可以利用 Tcl 脚本去检查设计的层次。通过观察设计的层次，工程师将选择那些独立于设计中的其它部分的层次化的模块来进行综合，这些在层次里的点作为编译的参考点。

一旦这些编译点选择好，然后把对时间的约束加到每一点；目前初始化这些点用手工的方法，在将来可以采用自动的。遵循着这些步骤，综合在设计中的较低层次的编译点上开始。当每一个编译点综合好以后，编译点的接口逻辑模块（ILM）会自动的产生，然后在较高的设计层次上做相同的工作。最终，顶层的设计用较低层的已产生的 ILM，和顶层的其他逻辑一块综合。这避免了花费大量的时间在写脚本上，而且避免了脚本错误，这使设计工程师们可以专注于提高设计的性能上。

ILM 是多点综合的核心技术。ILM 是局部的网表，能够从 Synplicity 或者第三方的产生网表的工具或综合工具中得到。它们是包含所有的边界信息和用于时序分析的逻辑的模型。因此与自顶向下的设计相比较，能够得到高质量的结果(QoR)。因为 ILM 仅仅包括端口到寄存器的逻辑，其他的部分都当作“黑盒子”，在综合一个大型的设计的时候能够大大的减少对内存的要求和运行的时间。一个大型模块的 ILM 比与它相关的整个设计的网表要小到 70%—80%。

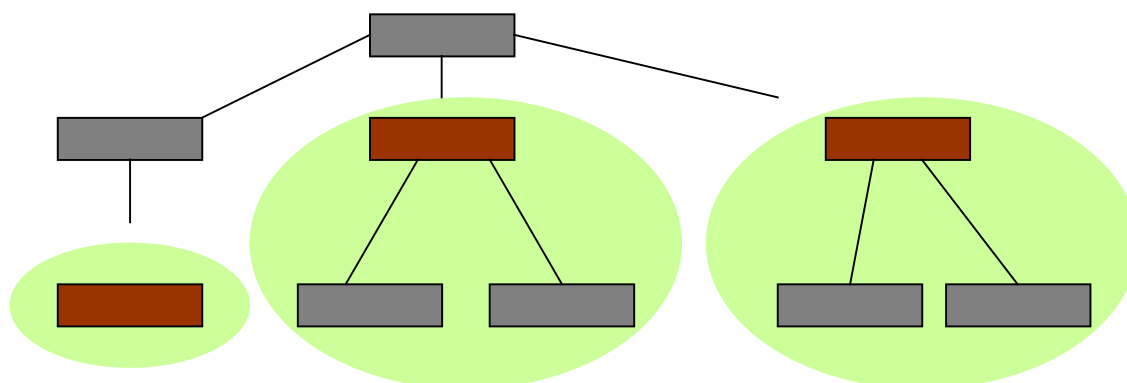
用户定义的编译点（表 1）是产生 ILM 的基础，也是多点综合技术的另一个关键的地方。编译点是独立综合的模块，综合后的结果用作包含它的模块和顶层的设计的综合。多点综合和其他的综合工具一个显著的区别是能够通过三种不同等级的编译点来控制边界的优化，三种编译点是：软编译点（soft compile point），硬编译点(hard compile point)和固定编译

点(locked compile point)。软编译点要做全边界的优化，意味着端口标志在综合的过程中会发生改变。相反，虽然包含在内部的和相邻的模块的边界做了优化，但是硬编译点的端口完整性得到了保留。对于固定边界点，在优化的过程中模块里的逻辑没有发生变化，虽然作为时序模型的 ILM 是产生了并且也许被优化了。因此，用户定义的编译点作为综合工具的指南，当模块化和综合设计的一个特定的部分时。通过 cross-boundary(交叉边界)优化，在自顶向下的设计中编译点是得到较高的 QoR 的关键因素。



设计的层次结构

表 1 另外一个多点综合技术的关键是采用用户定义的边界点，它是产生 ILM 的基础



使用编译点的层次化结构

通过用户定义的编译点和 ILM，多点综合技术支持 IP 的集成。多点综合技术自动的产生 IP 模型，并使用它的时序信息去做综合。把一个 IP 集成到整个的设计中，可对 IP 本身和它的相邻的模块做综合优化，并不会改变它的端口。当设计中有多个相同的 IP 时，边界优化可对其中一个特定的单元做优化，而不必对每一个 IP 都做优化。

多点综合技术应用一种独创的基于差别的增量综合处理方法 (Figure 3)，也就是在重新综合中仅仅综合真正有改变的模块。当改变 RTL 代码，约束条件后受到影响的编译点会被再综合。通过比较新的 RTL 代码和旧的 RTL 代码，多点综合技术能够智能化的确定那些改变需要重新综合。例如，改变 RTL 代码的时间标记，在文件中加上一些注释，重新安排约束条件的顺序，这些都不会激发综合工具做重新综合。这能够动态的提高综合效率，是更有效的综合方式。设计的稳定性借助于固定编译点得到了保障，使用户可以把需要做改变的模块隔离开来。

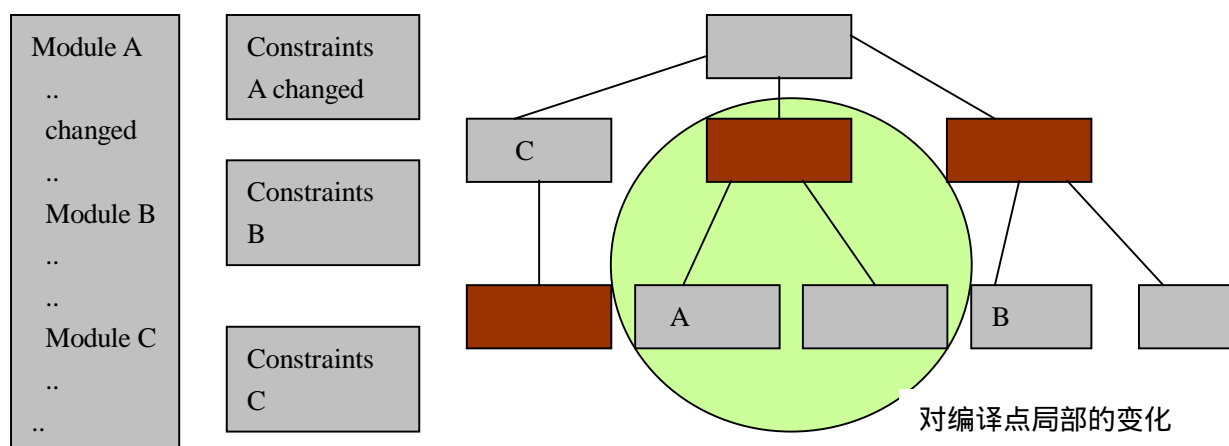


Figure 3:增量综合建立在实际的 RTL 级的代码，约束条件，和最小化设计的设计的变化与运行时间的基础上

多点综合技术同时可以应用于 ASIC 和 FPGA 的实现上，这在业界也是独一无二的。事实上，通过比较门数，嵌入的 IP 硬核，关键特性和为了实现设计所要花费的代价等方面，ASIC 和 FPGA 之间的界限变得模糊不清。随着产品的规模突破了大约十万个单元，ASIC 的设计需要有大的突破，越来越多的工程师见证了这两种选择之间的相互融合。同时对 ASIC 和 FPGA 的设计技术也将走向融合。以 Synplicity 公司的产品为例，物理综合，时序驱动的版图设计，形势验证不再仅仅是 ASIC 设计工程师所要考虑的事情。逐渐的，设计工程师不再仅仅依靠 ASIC 或者 FPGA 来实现他们的设计，甚至利用带有嵌入式 FPGA 和嵌入内核的 SoC 和 PsoC 来实现。那种能够以多种方式实现设计的 EDA 工具，例如多点综合技术，能够灵活的让你以最好的方式实现你的设计变，这一点是极其重要的。

多点综合技术为 FPGA 的设计提供了快速的增量设计流程

像先前所说的那样，多点综合技术可应用于 ASIC 和 FPGA 设计中，然而，对于 ASIC 和 FPGA 设计工程师来说，他们对工具的要求有微小的不同。ASIC 设计工程师需要高效的管理设计的门数，设计的划分和与综合相关的脚本文件，和对设计进行重新的划分，构建；而 FPGA 设计工程师需要快速的增量设计流程，并不需要考虑在设计的质量上折衷。多点综合技术用在 Altera 的逻辑设计和 Xilinx 的设计时，设计工程师在综合的过程中可以固定住设计的一部分不做改变。这将使已被验证的部分保持不变，仅需改变需要做出变化的部分。这种方法不仅保持了设计结果的可靠性，而且大大的降低了在设计发生变化后的综合和布局布线的时间。

使用多点综合技术

多点综合的流程很简单。第一步，工程师编译 HDL 代码，产生整个设计的 RTL 视图。接着，根据设计的关键的功能和关键的路径定义编译点。接下来，在整个的芯片上做自动的时序预估。一旦这些对时序的预估建立起来，然后每一个编译点根据它的时序预估做综合。自动的产生 ILM，并在顶层做时序分析和优化。通过同时在整个设计上为层次内的模块产生时序预估，多点综合技术产生初始化的时序预估。因此，不必要再花大量的时间去写约束

条件，进行时序预估的时间也比传统的方法快得多。

图 4a 是利用多点综合技术，在 0.11 微米的工艺上实现的一个 2M 门的设计。不像大多数的 SoC 设计那样，这个设计重复使用了一个 IP。这个 150K 的模块被复制了 9 次，因此可作为固定编译点。这个模块仅被映射一次，然后在顶层进行复制。在较低层的编译点被综合以后，只用被复制模块中的一个的 ILM 做自顶向下的综合，而其他的模块的逻辑保持不变，从而减少了 80% 的计算量。当与传统的自顶向下的综合方式（图 4b）相比时，我们发现内存的利用和花的时间减少了大约有 80%，而 QoR 却完全一样。

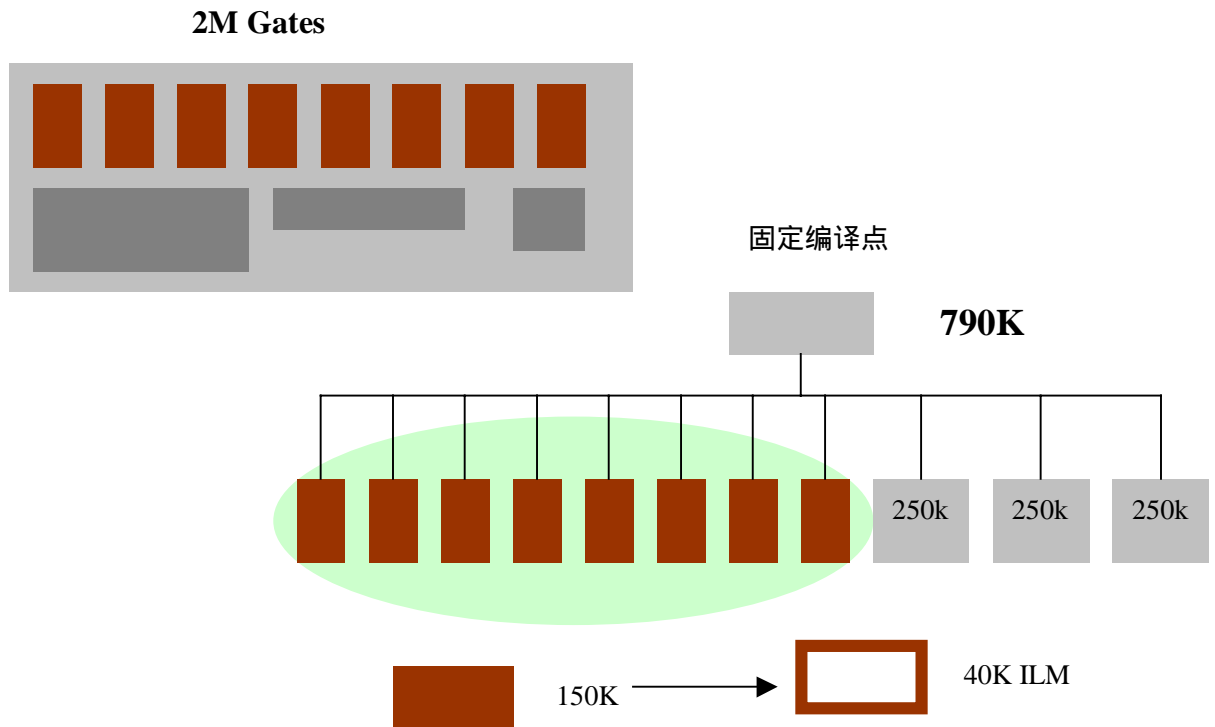


图 4a: 2M 门的 0.11 工艺基础上的设计内有已多次使用的 IP, 通过用较低层次上的 ILM 来代替同层次的 IP, 从而减少了上一层次的计算时间

	Memory Usage	Runtime	QoR
自顶向下的综合	3G	8 小时	满足 Timing
多点综合	.6G	1.5 小时	满足 Timing

图 4b: 与通常得自顶向下的设计方法相比较，多点综合技术在产生同样的 QoR 的同时，又最有效的内存利用率和运行时间

多点综合技术比传统的综合技术有很多的优势。不象传统的方法那样，一次只可综合 200K 门的设计，而多点综合技术一次却可以综合 1.5-2M 的设计。这就意味着设计工程师不必按工具所能处理的能力来划分设计，而可按照设计的功能和时序上的要求来划分。通过它的独一无二的增量设计能力的方式和可以自动的产生 ILM 和作时序上的预估，多点综合技

术提供了一种非常快的自动的解决方案。它的自顶向下的设计，层次化方法生成的 QoR 和自底向上的一样，而用固定编译点的增量综合保证了自底向上的设计的可靠性。在多点综合的流程中，ILM 作为在 IP 和大型模块之间的数据交换格式，与通常的传递大的脚本文件相比，传递编译点的文件要容易得多。

多点综合技术在 FPGA 和 ASIC 设计上的多功能性，使设计工程师不需要在掌握不同的设计工具上花大量的时间，工程师可以把主要的精力花在怎样最好的实现项目的设计上，无论是利用 FPGA，ASIC 或者是两者的结合。同样，多点综合技术的可扩充性，是在出现新的工艺技术的时候不必改变你的设计方法。最后一点，在业界，多点综合技术是为复杂 IC 综合提供解决方案的公认的领导者，这也决定了这个技术的可靠性和长久性。

未来的设计需要多点综合技术

多点综合技术也可适用于将来的设计，由于它的可扩充性。很多的设计工程师担心目前的综合方法不能适用于将来的设计上。考虑到这一点，Synplicity 公司开发了多点综合技术，使您在将来的设计不需要改变目前的综合方法。多点综合技术最基本的结构是可随着设计的需要集成更多的功能，能够支持将来的硬件，操作系统和内存的结构。多点综合技术也可以在多个微处理器上运行，每一个编译点在单独的微处理器上运行，最后整个的网表再集成在一起。

总结

经济和市场的压力促使我们在现有的技术上做出最好的设计。随着多点综合技术的出现，Synplicity 公司提供给设计工程师更先进的，更高能力的，高度自动化的设计流程。多点综合技术具有基于差别的增量综合技术，自动产生和使用 ILM,自定义的编译点等能力使这个设计流程在设计 FPGA 和 ASIC 的过程中比其它的方法更具有吸引力。多点综合技术包含了一个独一无二的传统的自顶向下和自底向上综合方法的结合和先进的综合技术，能够在今天设计高复杂的 SoC 和 PsoC 是保证它们的 QoR.